



跳点搜索融合双向并行蚁群算法的 AGV 路径规划研究

摘要

在静态栅格地图中,针对传统蚁群算法进行 AGV (Automated Guided Vehicle, 自动引导车) 路径规划收敛慢且搜索结果容易陷入局部最优的问题,提出一种融合跳点搜索 (Jump Point Search, JPS) 和双向并行蚁群搜索的改进算法。首先,对实际研究环境进行栅格化建模,使用改进的跳点搜索算法生成双向搜索的初始次优路径,为双向蚁群搜索提供初始搜索方向参考。其次,在双向并行蚁群搜索过程中采用改进的转移概率启发函数,该函数在确定下一个转移节点时考虑了避免 AGV 与障碍物碰撞的因素,同时通过设计信息素共享机制并结合改进的信息素增量及浓度两种融合模型,共享和更新全局信息素浓度,以更好地探索和优化路径,保证双向路径连结。最后,与传统蚁群算法进行实验结果对比,验证了改进算法的全局搜索能力、效率和安全性。

关键词

跳点搜索算法;蚁群算法;自动引导车;路径规划;双向并行

中图分类号 TP18;TP242.6

文献标志码 A

收稿日期 2023-10-23

资助项目 福建省自然科学基金(2021J01331)

作者简介

林信川,男,副教授,研究方向为智能信息处理. aikey.lin@foxmail.com

0 引言

AGV 路径规划 (AGVPP, AGV Path Planning) 是指为每台 AGV (Automated Guided Vehicle, 自动引导车) 设备规划一条自起点至终点且规避与障碍物碰撞的最优路径^[1]。AGV 路径规划目前主要有三类方法: 1) 基于人工智能的算法, 如蚁群算法^[2]、粒子群优化算法^[3]等; 2) 基于局部避障的算法, 如势场法^[4]、动态窗口法^[5]等; 3) 基于图论的算法, 如 Dijkstra 算法^[6]、A* 算法^[7]等。基于人工智能的算法可根据环境变化自适应地调整路径规划策略, 适用于不同的场景和任务, 同时可能伴随大量的计算和模拟, 需要较高的计算资源和时间成本。基于局部避障的算法只考虑 AGV 周围的局部环境, 而不需要知道全局地图信息, 具有实时性强、精度高和适应性强的特点, 但可能会导致生成的路径仅为局部最优解, 而非全局最优解。基于图论的算法能够基于全局环境信息进行搜索和规划, 即使在复杂的环境中也能够规划一条连接起点和终点的路径, 但有可能存在较大的计算量, 需通过优化策略来提升规划路径的质量和效率。

A* 算法是静态场景下进行 AGV 路径规划的常见启发式算法, 但在搜索路径的时候, 需要频繁查询 OpenList 和 CloseList 中的节点, 导致节点排序和查询节点消耗大量时间资源, 随着栅格地图的增大, AGV 的导航实时性将明显降低。Harabor 等^[8]提出了跳点搜索 (Jump Point Search, JPS) 算法, 只将所谓的跳跃点作为搜索的节点, 不仅显著提升了搜索效率, 也在一定程度上解决了因对称性路径造成的搜索空间过大的问题。赵晓等^[9]将 JPS 算法应用于移动机器人的导航中, 体现了 JPS 算法较传统 A* 算法的优势。但随着地图规模和障碍物复杂度的增加, JPS 算法的路径规划过程仍可能导致计算资源的大量消耗, 同时考虑到障碍物的实物体积, 算法生成的规划路径无法有效避免 AGV 与障碍物碰撞, 可能存在紧贴障碍物、轨迹不平滑^[10]等问题, 将直接影响实际工作场景下 AGV 设备的安全运行。

传统蚁群算法是一种基于正反馈的单向搜索算法, 其前期搜索的盲目性有可能导致收敛速度慢且容易陷入局部最优^[11], 但其搜索速度快、精度高、寻址能力强, 同时易与其他算法结合, 因此具有较高的优化灵活性。马军等^[12]尝试将蚁群算法和 A* 算法进行结合, 改进了启发式函数和信息素更新方式, 但并没有显著解决蚁群算法搜索初期盲目性大的问题。Zhu 等^[13]融合了人工势场法与蚁群算法, 通过

¹ 福建商学院 信息工程学院, 福州, 350012

动态调整算法的状态转移规则以提升全局搜索能力和收敛进度,但算法复杂度对 AGV 配置有一定要求,且存在 AGV 碰撞障碍物边缘的可能性.李二超等^[14]提出一种基于贝塞尔曲线融合双向蚁群的算法,规划出的路径可避免 AGV 与障碍物的碰撞且兼顾了平滑度的要求,但是对搜索初期如何优化双向蚁群搜索的方向性未做探讨,且算法的复杂性和计算量较大,对 AGV 的性能配置存在一定要求.

针对上述问题,本文提出一种利用跳点搜索算法加速双向并行蚁群搜索的改进算法,两组蚁群自地图起点和终点并行相向搜索,利用跳点搜索算法规划的次优路径为两组蚁群提供初始路径参考,以加速搜索和算法收敛,并通过在并行搜索的蚁群间共享信息素增量和信息素浓度,以进一步优化搜索效果.

1 环境建模

栅格法建模具有简单直观、易于计算和适用性广等优点,是路径规划中常见的建模方法^[15].本文采用栅格法对实际工作环境展开建模研究.模型环境中每个栅格为一个节点,黑色节点表示障碍物,白色节点表示 AGV 可通过的空旷区域,地图信息即可表示为一个二维矩阵 $C = (c_{ij})$,其中:

$$c_{ij} = \begin{cases} 0, & \text{空旷区域;} \\ 1, & \text{障碍物.} \end{cases} \quad (1)$$

假设 AGV 每次可以进行水平、垂直或斜向 45° 共 8 个方向的移动,每次移动距离为 1 或 $\sqrt{2}$ 单位,本

文研究过程中 1 个距离单位为 1 m.

为验证算法的通用性和可用性,本文采用栅格法构建如下 $20\text{ m} \times 20\text{ m}$ 和 $30\text{ m} \times 30\text{ m}$ 两个不同规模的模型地图,如图 1 所示.

上述模型地图起点和终点分别使用 S 、 E 表示,本文算法研究的目的可归纳为寻找 S 和 E 间的最短平滑安全路径问题.根据算法验证需要,模型地图中的部分障碍物为“凹”形并由多个障碍物节点组成,其他障碍物在实际生产环境中表现为随机形状和随机分布.

2 跳点搜索算法及其不足

2.1 跳点搜索算法

算法搜索过程中当前节点的邻节点,可分为自然邻居和强迫邻居^[16]两类.自然邻居节点是当前节点在水平、垂直和对角线方向上的直接邻节点,用于进一步扩展路径搜索.在上述研究假设的 AGV 运动方向上,若当前节点 c 的邻节点中有障碍物,相较其他邻节点,邻节点 f 到节点 c 的父节点 p 的代价距离最短,则节点 f 为节点 c 的强迫邻居节点.若节点 c 满足下列条件之一,则可认定为跳点:

- 1) 节点 c 是起点或终点;
- 2) 节点 c 至少有一个强迫邻居;
- 3) 在斜向搜索时,斜向搜索分解的水平或垂直方向上,有满足上述 1) 或 2) 条件的节点.

在搜索过程中算法通过跳点剔除搜索过程中大量无用节点,并反复通过代价估计函数值确定下一

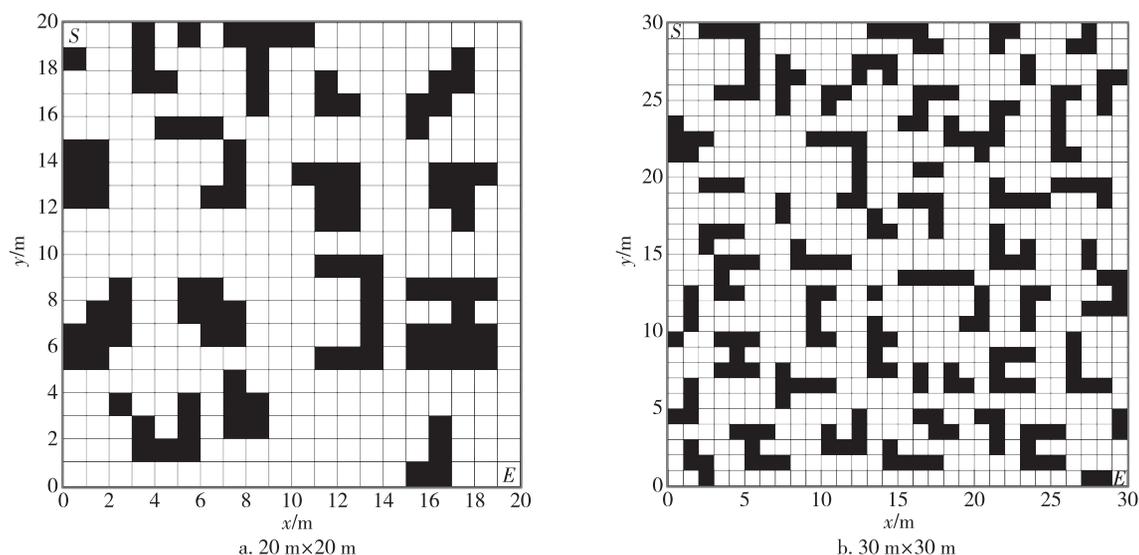


图 1 栅格地图

Fig. 1 Grid maps

步需拓展的节点,跳点搜索算法的代价估计函数与 A* 算法相同,定义为

$$f(n) = g(n) + h(n). \quad (2)$$

其中: $f(n)$ 为节点 n 到目标点的代价估计函数; $g(n)$ 为实际代价函数,反映了自起点到节点 n 的实际代价; $h(n)$ 为预计代价启发函数,估计了自节点 n 至终点的代价,通常采用欧式距离^[17]、曼哈顿距离^[18]或切比雪夫距离^[19]进行计算.在跳点搜索过程中优先考虑 $f(n)$ 值更小的节点,从而尽可能地减少搜索空间,提高寻路效率.

2.2 跳点搜索算法的不足

跳点搜索算法基于 A* 算法进行改良,在路径搜索中舍弃了大量可能会产生冗余计算的无用节点,仅识别和存储“跳点”,相较于 A* 算法显著减少了算法运行期间需要计算的节点数量和探索路径数量.但跳点搜索算法在路径规划时并不考虑 AGV 及障碍物的形状和大小,只关注节点间的跳跃关系,因此规划出来的路径有可能存在 AGV 紧贴障碍物或与之碰撞的问题^[20].当障碍物形状呈现对称图形时,若双向搜索均使用跳点算法,则可能生成从障碍物对称侧经过障碍物的两条长度和收敛速度类似的路径;若双向搜索未进行信息共享,则一向路径进行跳点扩展时,有可能越过另一向路径正在扩展的跳点,进而产生冗余路径.在障碍物位置分布随机的情况下,若障碍物数量较多,路径规划过程中有可能生成大量跳点,造成搜索方向和路径长度存在较大的不确定性,且双向两条搜索路径的重叠节点有可能距离起点和终点连线的中心较远,造成路径规划资源浪费.

3 融合跳点搜索的双向并行蚁群算法

传统蚁群算法模拟一组蚂蚁多次迭代搜索食物的过程,即自起点至终点的单向搜索,其转移概率可以表示为

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{l \in J^k} [\tau_{il}(t)]^\alpha [\eta_{il}(t)]^\beta}, & j \in J^k; \\ 0, & j \notin J^k. \end{cases} \quad (3)$$

其中: p_{ij}^k 表示第 k 只蚂蚁在节点 i 时选择连接节点 i 和节点 j 的路径的概率; $\tau_{ij}(t)$ 表示节点 i 和节点 j 之间的信息素浓度; $\eta_{ij}(t)$ 表示节点 i 和节点 j 之间的启发函数值; α 为信息素浓度控制参数, β 为启发函数参数,分别控制信息素浓度和启发函数对路径选择的影响; J^k 表示第 k 只蚂蚁可以选择的路径集合.

考虑到传统蚁群算法前期搜索的盲目性和单向

搜索效率,本文提出一种双向并行搜索的蚁群算法,即位于起点和终点的两组蚂蚁同时进行并行搜索.在搜索的起始阶段,将改进的跳点搜索算法的启发式规则和路径作为蚁群算法的启发式信息和初始次优路径.在搜索过程中,双向蚁群通过共享和同步更新信息素增量和信息素浓度,达到优化搜索的方向性的目的.当两组蚁群搜索路径相接并重叠时,重叠节点的信息素浓度将会显著升高,即重叠节点可以帮助确定搜索方向的正确性,并提供路径继续搜索的方向性参考.两组蚁群分别到达目标点后,通过比较两个方向上的路径质量评价指标来选择最优路径.

以下研究将起点 S 至终点 E 的路径搜索方向定为正向搜索方向,将终点 E 至起点 S 的路径搜索方向定为反向搜索方向.

3.1 跳点搜索算法改进

传统蚁群算法要求单向蚁群自起点到目标点进行探索,所有蚂蚁在搜索过程中均会持续释放信息素,并同时伴随信息素的挥发效应.单向蚁群大致相同的起始路径将无法保证算法有效规划全局最优路径,且均匀分配的信息素浓度虽然保证了所有路径被选择的概率,但也随之产生了蚁群初期盲目搜索及引发的算法搜索效率降低的问题.

为提升路径规划效率、降低计算量,本研究拟在搜索初期使用改进的跳点搜索算法生成初始次优路径,将路径方向作为蚁群算法中的路径选择策略.同时,本文将采用曼哈顿距离作为跳点搜索的启发函数.

考虑到跳点搜索算法规划的路径有可能造成 AGV 与障碍物碰撞,对跳点筛选规则做如下改进:

1) 直线搜索方向筛选规则改进

直线搜索包括水平和垂直两个方向,两个方向的改进原理相同.以水平方向搜索为例,改进的直线搜索路径如图 2a 中的实线路径所示. $p(c)$ 为节点 c 的父节点,节点 f 为节点 c 的强制邻居,当节点 c 的邻节点存在障碍物时(图中为节点 c 的正上方黑色节点),为避免碰撞,本研究不允许通过虚线对角线到达强制邻居,即在图中不以节点 c 作为跳点,而以其自然邻居节点 d 作为跳点.

2) 斜线搜索方向筛选规则改进

改进的对角线搜索路径规则如图 2b 中的实线路径所示,节点 c 的左侧为障碍物.与改进的直线搜索类似,为避免碰撞则不以虚线路径行进,同理将节

点 b 和节点 d 作为跳点处理.

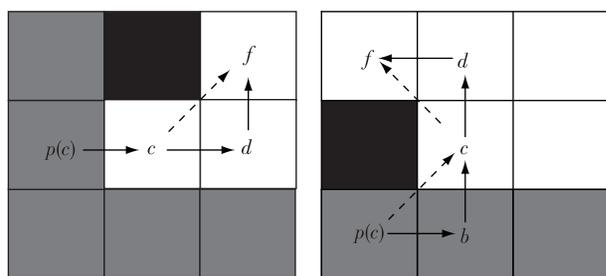


图 2 跳点搜索规则改进

Fig. 2 Improvement of jump point search rules

3.2 转移概率启发函数改进

传统蚁群算法在 t 时刻的转移概率启发函数一般表示为 $\eta_{ij}(t) = \frac{1}{d_{ij}}$, 启发函数仅考虑节点 i 和 j 之间的欧式距离 d_{ij} , 即 i 节点附近节点的局部信息. 但某些情况下, 节点 i 和 j 之间的欧式距离与节点 j 到目标点的距离未必成正比.

改进后的转移概率启发函数综合考虑了局部及全局信息, 同时结合了上述对跳点筛选规则的优化思路, 使用曼哈顿距离作为转移概率启发函数, 符合上述改进跳点筛选规则避免 AGV 与障碍物碰撞的思路. 改进后的启发函数为

$$\eta_{ij}(t) = \frac{1}{d_{iT}}, \quad (4)$$

$$d_{iT} = \begin{cases} |i_x - E_x| + |i_y - E_y|, & \text{正向搜索,} \\ |i_x - S_x| + |i_y - S_y|, & \text{反向搜索.} \end{cases}$$

其中: d_{iT} 为节点 i 到目标点 T 的曼哈顿距离, 可以简化计算、提升效率和避免碰撞; 考虑到双向搜索, 目标点 T 可能为起点 S 或终点 E ; $i_x, i_y, S_x, S_y, E_x, E_y$ 分别为节点 i 、起点 S 和终点 E 的横坐标及纵坐标.

3.3 信息素共享机制设计及更新公式改进

3.3.1 传统蚁群算法信息素更新机制

传统蚁群算法的信息素浓度更新公式表示为

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^n \Delta\tau_{ij}^k(t). \quad (5)$$

其中: $\tau_{ij}(t)$ 表示在时刻 t 时节点 i 和节点 j 之间的信息素浓度; ρ 为信息素挥发因子, 取值范围为 $(0, 1)$, 表示信息素的挥发速度; $\Delta\tau_{ij}^k(t)$ 表示第 k 只蚂蚁在节点 i 到节点 j 间留下的信息素增量. 在信息素更新过程中浓度较高的路径会被更多的蚂蚁选择, 从而留下更多的信息素, 进一步增强其信息素浓度. 在计算 $\Delta\tau_{ij}^k(t)$ 时, 可以根据蚂蚁经过路径的质量和长度

来计算:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & (i, j) \in T_k; \\ 0, & \text{否则.} \end{cases} \quad (6)$$

其中: Q 为信息素常数; L_k 表示第 k 只蚂蚁经过路径的长度; T_k 表示第 k 只蚂蚁经过的路径集合.

3.3.2 信息素共享机制设计

式(5)和(6)主要应用于蚁群的单向搜索过程, 传统蚁群算法在选择路径时, 可能会面临大量的盲路径, 经过一定数量路径的迭代后生成的可能为非最优路径, 导致计算资源浪费. 本文在双向并行蚁群搜索过程中, 拟让双向蚁群共享对方蚁群全局信息素信息, 即当一方的蚂蚁发现了一条可行的路径时释放信息素, 另一方的蚂蚁可以通过感知到的全局信息素来引导搜索方向, 使两个方向的搜索路径逐渐靠近, 确保路径连结, 提升搜索效率. 全局信息素信息共享可以增进蚁群间的信息交流和合作, 有助于更好地探索和优化路径.

本研究中的全局信息素信息共享分为信息素增量共享和信息素浓度共享. 对以下研究作假设如下: 当前时刻 t 正向最近完成的搜索路径为从节点 i 至节点 j , 反向为节点 l 至节点 k . 需要注意的是, 为保证双向蚁群并行搜索效率, 若当前此轮正向搜索结束但此时反向搜索仍在进行, 则增量融合公式中将使用反向搜索上一时刻生成的最新增量, 即双向搜索过程共享最新的增量矩阵信息, 反向搜索同理.

3.3.3 信息素增量融合模型设计

首先针对传统蚁群算法的信息素增量公式进行改进, 设计双向信息素增量融合模型. 在每轮搜索迭代结束时, 通过共享和融合双向信息素增量, 生成当前搜索方向的信息素增量, 并将对向搜索的信息素增量作为生成当前信息素增量的参考, 以此作为引导双向蚁群路径连结的激励.

首先对式(6)进行改进, 简化计算并融入对预计代价的考量, 将基础增量公式设计如下:

正向搜索:

$$\Delta\tau_{ij}^k S(t) = \begin{cases} \frac{Q}{d_{iE}}, & (i, j) \in T_k; \\ 0, & \text{否则.} \end{cases} \quad (7)$$

反向搜索:

$$\Delta\tau_{lk}^k E(t) = \begin{cases} \frac{Q}{d_{lS}}, & (k, l) \in T_k; \\ 0, & \text{否则.} \end{cases} \quad (8)$$

其中: $\Delta\tau_{ij}^+S(t)$ 和 $\Delta\tau_{lk}^-E(t)$ 分别表示在 t 时刻的正向和反向的基础增量; d_{iE} 和 d_{lS} 分别表示当前搜索节点 i 或 l 至终点 E 或起点 S 的曼哈顿距离.

在双向基础增量的基础上进一步进行增量融合,信息素增量融合模型可表示为

$$\begin{cases} \Delta\tau_{ij}^+S(t) = (1 - \varepsilon) \Delta\tau_{ij}^+S(t) + \varepsilon \Delta\tau_{lk}^-E(t), \\ \text{正向搜索;} \\ \Delta\tau_{lk}^-E(t) = (1 - \delta) \Delta\tau_{lk}^-E(t) + \delta \Delta\tau_{ij}^+S(t), \\ \text{反向搜索.} \end{cases} \quad (9)$$

其中: $\Delta\tau_{ij}^+S(t)$ 及 $\Delta\tau_{lk}^-E(t)$ 分别表示正向和反向搜索在 t 时刻的融合增量; ε 和 δ 为取值 $(0,1)$ 范围的权重系数,分别用于控制正向和反向信息素增量的相对贡献; Q 为信息素常数,含义与传统蚁群算法更新公式相同.

3.3.4 信息素浓度融合模型设计

参考基础增量和融合增量进行双向信息素浓度融合模型设计.与式(5)类似,利用式(9)中得到的双向融合增量,分别进行相应方向信息素基础浓度值更新,更新过程已将对向搜索的信息素增量考虑在内.更新公式可表示为

$$\begin{cases} \tau_{ij}^+S(t+1) = (1 - \rho) \tau_{ij}^+S(t) + \Delta\tau_{ij}^+S(t), \\ \tau_{lk}^-E(t+1) = (1 - \rho) \tau_{lk}^-E(t) + \Delta\tau_{lk}^-E(t). \end{cases} \quad (10)$$

其中: $\tau_{ij}^+S(t)$ 和 $\tau_{lk}^-E(t)$ 分别表示正向搜索和反向搜索在 t 时刻的基础信息素浓度; ρ 为信息素挥发因子,与传统蚁群算法相同,取值范围为 $(0,1)$,表示信息素的挥发速度.

接下来根据得到的双向基础信息素浓度值,融合生成当前搜索方向的融合信息素浓度.与信息素增量融合模型类似,信息素浓度融合模型可表示为

$$\begin{cases} \tau_{ij}^+S(t) = (1 - \mu) \tau_{ij}^+S(t) + \mu \tau_{lk}^-E(t), \\ \tau_{lk}^-E(t) = (1 - \varphi) \tau_{lk}^-E(t) + \varphi \tau_{ij}^+S(t). \end{cases} \quad (11)$$

其中: $\tau_{ij}^+S(t)$ 和 $\tau_{lk}^-E(t)$ 分别表示正向和反向的融合信息素浓度; μ 和 φ 为取值 $(0,1)$ 范围的权重系数,分别用于控制正向和反向上信息素浓度的相对贡献.

3.4 改进算法流程

改进算法的流程如图3所示.

在搜索过程开始阶段,首先针对实际研究环境进行栅格化建模,存储起点 S 、终点 E 和障碍物的分布坐标信息,同时针对关键参数进行初始化.使用改进的跳点搜索算法生成双向搜索的初始次优路径,该路径作为双向搜索过程开始时更新初始信息素含量的依据,为双向蚁群搜索提供方向参考.在初始次

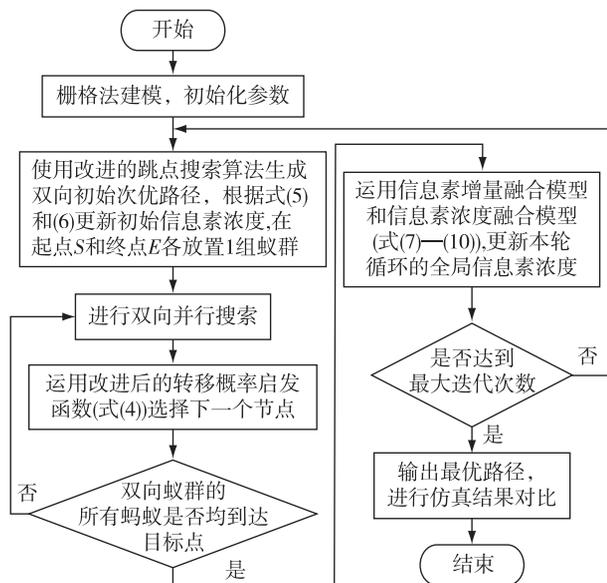


图3 改进算法流程

Fig. 3 Flow chart of improved algorithm

优路径的基础上生成初始信息素浓度,并展开双向并行搜索.搜索过程将采用改进的转移概率启发函数,该函数在确定下一个转移节点时已考虑了避免AGV与障碍物碰撞的可能性,因此转移节点可能为信息素浓度最高或次高的节点.通过共享双向信息素增量和浓度值,使用信息素增量融合模型和浓度融合模型更新全局信息素浓度.当达到最大迭代次数时输出最优路径,并与传统蚁群算法进行实验结果对比.

4 实验及结果分析

为验证本文算法的有效性,采用 Python 3.7 进行实验.实验环境为 16 GB 内存运行 Windows 11 的 PC, CPU 为具备 4 核 8 线程的 AMD Ryzen 5,工作频率 2.1 GHz.实际工作环境为某电子产品制造企业已普遍采用 AGV 进行产品及物料搬运的两个车间,根据障碍物分布栅格化 $20\text{ m} \times 20\text{ m}$ 和 $30\text{ m} \times 30\text{ m}$ 两组地图环境,建立直角坐标系进行实验.出于隐私保护考虑,AGV 设备型号和非关键参数在不影响实验研究效果的前提下不作体现.本文开展实验车间使用 4 轮额定功率 80 W 的 AGV 设备,设备长宽高尺寸为 $0.8\text{ m} \times 0.6\text{ m} \times 0.4\text{ m}$,理想巡航速度和最大速度分别为 0.5 m/s 和 1 m/s ,加速度和减速度均为 1 m/s^2 .

实验过程中设定两组蚂蚁种群数量均为 $m = 30$,最大迭代次数 $N = 200$,其他参数为 $\alpha = 2, \beta = 5, \rho = 0.55, Q = 2, \varepsilon = 0.45, \delta = 0.5, \mu = 0.35, \varphi =$

0.27, 以下结果为每组地图分别模拟算法 200 次的平均值. 具体实验结果如图 4 和图 5 所示.

将图 4 和图 5 的实验数据进行进一步对比, 分别如表 1 和表 2 所示.

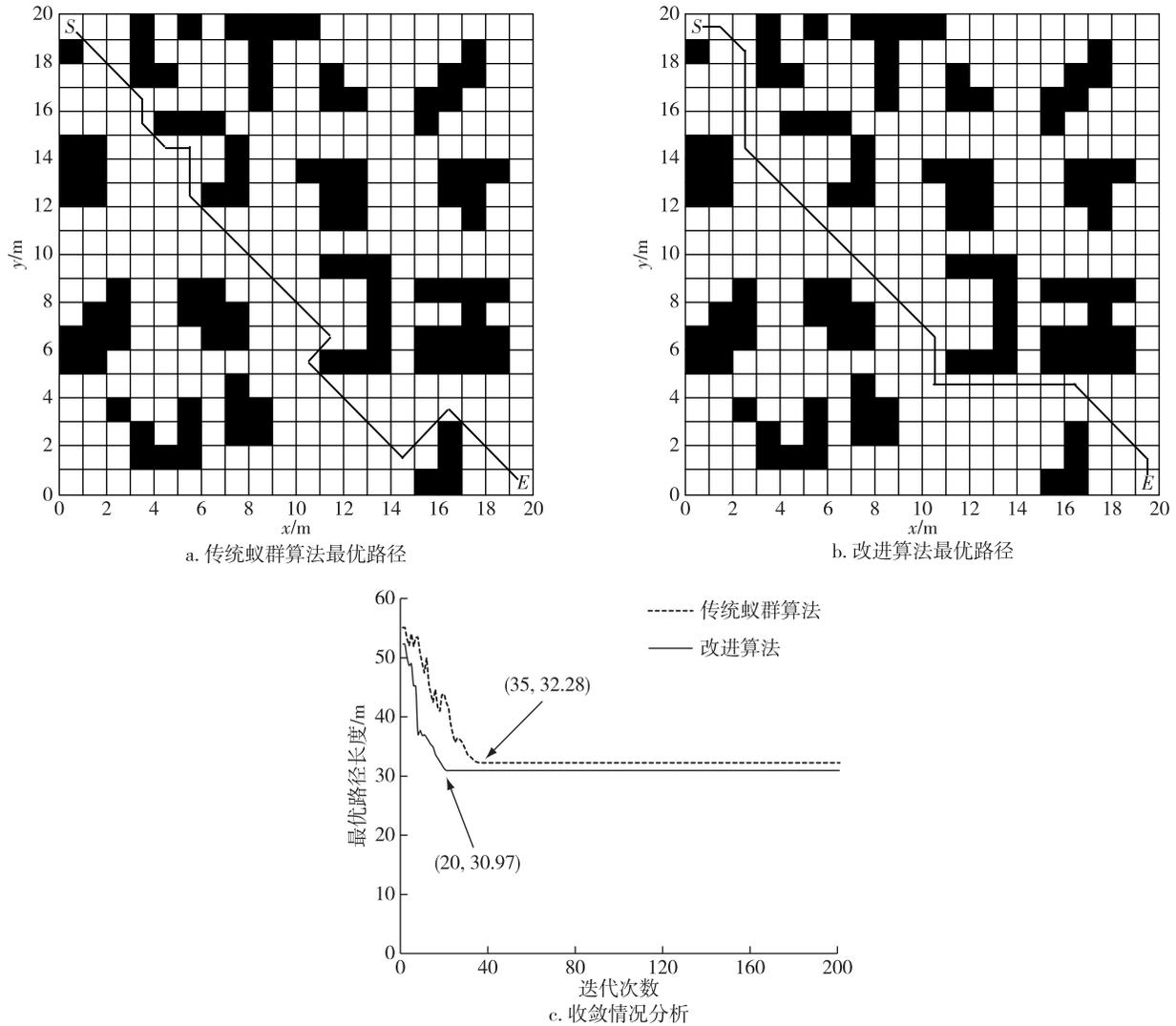


图 4 20 m×20 m 环境实验分析

Fig. 4 Environment experimental analysis at 20 m×20 m

表 1 20 m×20 m 环境下实验数据对比

Table 1 Comparison of experimental data under 20 m×20 m environment

算法	路径长度/m				拐点数				迭代次数均值	搜索耗时均值/s	搜索节点均值
	最优值	最大值	平均值	标准差	最优值	最大值	平均值	标准差			
传统蚁群算法	32.28	55.15	37.34	3.57	9	25	14.07	3.19	41.75	24.31	71.79
改进算法	30.97	52.30	32.01	0.19	7	9	7.97	0.37	23.50	13.53	37.50

表 2 30 m×30 m 环境下实验数据对比

Table 2 Comparison of experimental data under 30 m×30 m environment

算法	路径长度/m				拐点数				迭代次数均值	搜索耗时均值/s	搜索节点均值
	最优值	最大值	平均值	标准差	最优值	最大值	平均值	标准差			
传统蚁群算法	47.59	99.95	58.29	6.51	17	57	37.90	8.31	112.12	50.12	121.73
改进算法	45.11	89.30	47.40	0.21	11	13	12.93	0.25	35.90	25.11	58.50

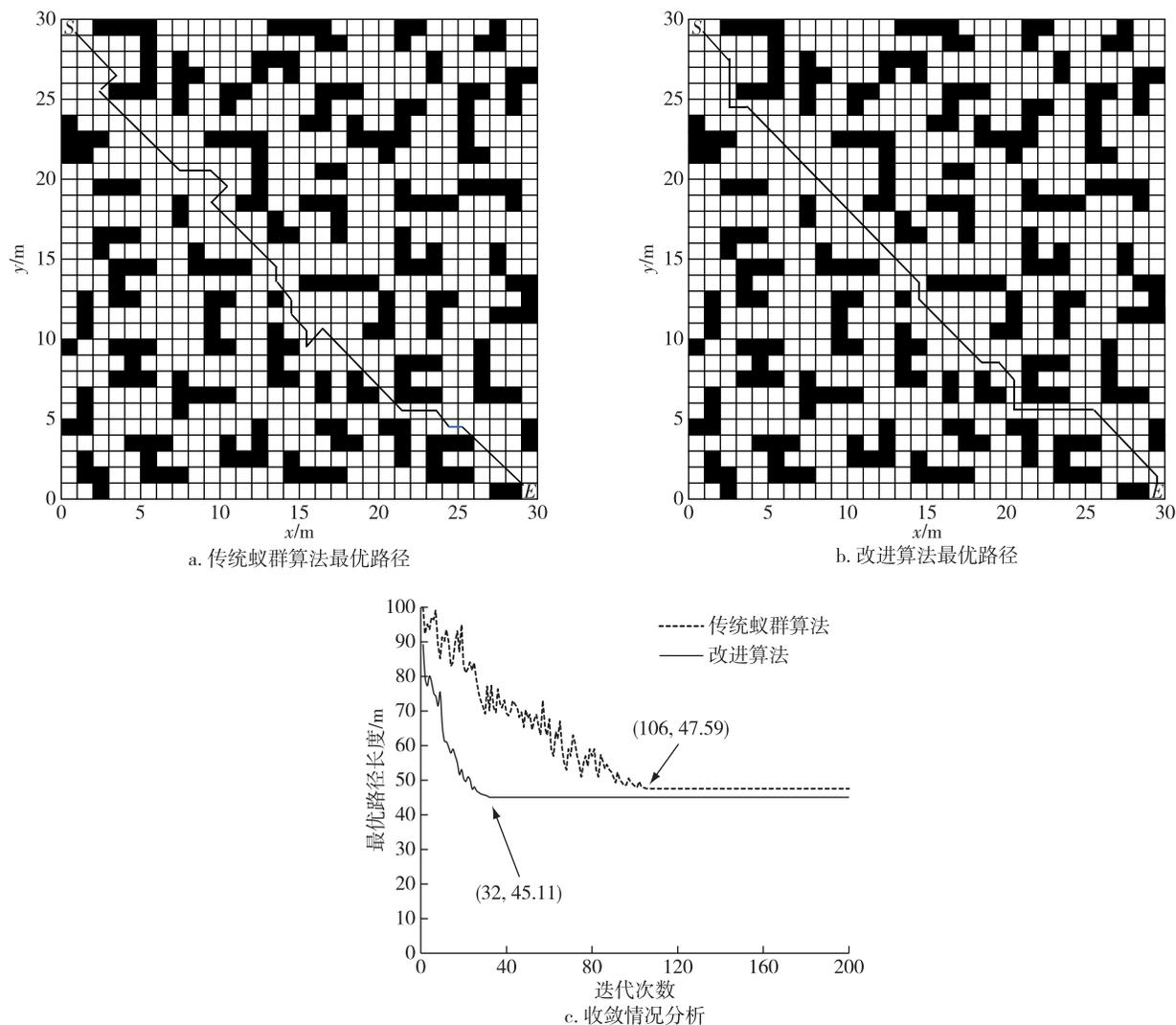


图 5 30 m×30 m 环境实验分析

Fig. 5 Environment experimental analysis at 30 m×30 m

通过分析表 1 和表 2 对比数据,传统蚁群算法在搜索初期由于搜索的盲目性及其导致的无用信息素积累,有可能在探索过程生成冗余拐点,增加最优路径长度、拐点数和搜索耗时.改进算法能够有效减少搜索节点的个数,有助于减少规划过程对系统资源的占用,提升规划效率.从收敛曲线可知,虽然改进算法在搜索初期有所波动,但生成其最优路径的迭代次数显著减少,收敛速度快,结合相关对比指标的平均值和标准差可知,改进算法的平滑性明显优于传统蚁群算法.综上,改进算法各项指标均优于传统蚁群算法,具有较好的全局搜索能力,且伴随着地形扩大和障碍物增加,改进算法与传统蚁群算法的指标差距在逐步拉大,在实际环境中有助于规划出更加平滑的路径,减少 AGV 工作过程的时间和能耗.同时,将改进算法中的跳点搜索规则融入转移概

率启发函数,生成的路径能够有效地避免与障碍物碰撞,大大提升了规划路径的安全性.

此外,本文研究并未对算法适用环境做明确限制,也未固化障碍物的位置,具备实际工作场景下的应用价值.

5 结语

针对传统蚁群搜索算法收敛速度慢、规划路径易陷入局部最优且未考虑 AGV 与障碍物的碰撞的不足,本文提出一种将跳点搜索算法融入双向并行蚁群搜索过程的改进算法.改进算法设定了两组相向并行搜索的蚁群,以改进的跳点搜索算法为双向搜索生成初始次优路径,以此减少蚁群搜索初期搜索的盲目性.在双向并行搜索过程中通过改进转移概率启发函数、共享信息素增量和浓度优化了全局

搜索能力、效率和安全性。通过多次迭代实验并对比数据,改进算法较传统蚁群算法在复杂障碍物环境中,具有更快的收敛速度、更短的规划路径和耗时以及更少的搜索节点,验证了本文改进算法的有效性和优越性。

参考文献

References

- [1] Patle B K, Babu L G, Pandey A, et al. A review: on path planning strategies for navigation of mobile robot[J]. Defence Technology, 2019, 15(4): 582-606
- [2] 刘加奇, 王泰华, 董征. 基于改进蚁群算法的机器人路径规划[J]. 传感器与微系统, 2022, 41(5): 140-143
LIU Jiaqi, WANG Taihua, DONG Zheng. Path planning of mobile robot based on improved ant colony algorithm[J]. Transducer and Microsystem Technologies, 2022, 41(5): 140-143
- [3] 赵迪, 何克勤, 赵祖高. 基于改进粒子群优化算法的机器人路径规划[J]. 传感器与微系统, 2023, 42(6): 150-153
ZHAO Di, HE Keqin, ZHAO Zugao. Path planning for mobile robot based on improved PSO algorithm[J]. Transducer and Microsystem Technologies, 2023, 42(6): 150-153
- [4] 翟丽, 张雪莹, 张闲, 等. 基于势场法的无人车局部动态避障路径规划算法[J]. 北京理工大学学报, 2022, 42(7): 696-705
ZHAI Li, ZHANG Xueying, ZHANG Xian, et al. Local dynamic obstacle avoidance path planning algorithm for unmanned vehicles based on potential field method[J]. Transactions of Beijing Institute of Technology, 2022, 42(7): 696-705
- [5] 陈娇, 徐菱, 陈佳, 等. 改进 A* 和动态窗口法的机器人路径规划[J]. 计算机集成制造系统, 2022, 28(6): 1650-1658
CHEN Jiao, XU Ling, CHEN Jia, et al. Path planning based on improved A* and dynamic window approach for mobile robot[J]. Computer Integrated Manufacturing Systems, 2022, 28(6): 1650-1658
- [6] 吴立辉, 胡文博, 周秀, 等. 面向多搬运任务的柔性制造车间多载具 AGV 节能路径规划[J]. 计算机应用研究, 2023, 40(1): 57-62
WU Lihui, HU Wenbo, ZHOU Xiu, et al. Energy-efficient path planning for single multi-load AGV executing multiple transport tasks in flexible manufacturing workshop[J]. Application Research of Computers, 2023, 40(1): 57-62
- [7] 耿宏飞, 神健杰. A* 算法在 AGV 路径规划上的改进与验证[J]. 计算机应用与软件, 2022, 39(1): 282-286
GENG Hongfei, SHEN Jianjie. Improvement and verification of A* algorithm in AGV path planning[J]. Computer Applications and Software, 2022, 39(1): 282-286
- [8] Harabor D, Grastien A. The JPS pathfinding system[J]. Proceedings of the International Symposium on Combinatorial Search, 2012, 3(1): 207-208
- [9] 赵晓, 王铮, 黄程侃, 等. 基于改进 A* 算法的机器人路径规划[J]. 机器人, 2018, 40(6): 903-910
ZHAO Xiao, WANG Zheng, HUANG Chengkan, et al. Mobile robot path planning based on an improved A* algorithm[J]. Robot, 2018, 40(6): 903-910
- [10] 魏博闻, 严华. 一种面向非结构化环境的改进跳点搜索路径规划算法[J]. 科学技术与工程, 2021, 21(6): 2363-2370
WEI Bowen, YAN Hua. An improved jump point search path-planning algorithm for unstructured environment[J]. Science Technology and Engineering, 2021, 21(6): 2363-2370
- [11] 刘紫玉, 赵丽霞, 薛建越, 等. 面向车辆路径问题的改进蚁群算法研究[J]. 河北科技大学学报, 2022, 43(1): 80-89
LIU Ziyu, ZHAO Lixia, XUE Jianyue, et al. Research on vehicle routing problem based on improved ant colony algorithm[J]. Journal of Hebei University of Science and Technology, 2022, 43(1): 80-89
- [12] 马军, 宋栓军, 韩军政, 等. 融合蚁群-A* 算法的机器人路径规划[J]. 西安工程大学学报, 2020, 34(1): 72-77
MA Jun, SONG Shuanjun, HAN Junzheng, et al. Mobile robot path planning based on a fusion ant colony-A* algorithm[J]. Journal of Xi'an Polytechnic University, 2020, 34(1): 72-77
- [13] Zhu S N, Zhu W Y, Zhang X Q, et al. Path planning of lunar robot based on dynamic adaptive ant colony algorithm and obstacle avoidance[J]. International Journal of Advanced Robotic Systems, 2020, 17(3): 172988141989897
- [14] 李二超, 齐款款. 贝塞尔曲线融合双向蚁群算法的机器人路径规划[J]. 陕西师范大学学报(自然科学版), 2022, 50(3): 79-88
LI Erchao, QI Kuankuan. Path planning of mobile robot based on Bessel curve and bidirectional ant colony algorithm[J]. Journal of Shaanxi Normal University (Natural Science Edition), 2022, 50(3): 79-88
- [15] 付雄, 李涛. 基于自适应步长策略的 A* 算法路径规划优化[J]. 南京信息工程大学学报(自然科学版), 2024, 16(2): 164-172
FU Xiong, LI Tao. Path optimization of A* algorithm based on adaptive step size strategy[J]. Journal of Nanjing University of Information Science & Technology (Natural Science Edition), 2024, 16(2): 164-172
- [16] 邱磊, 刘辉玲, 雷建龙. 跳点搜索算法的原理解释及性能分析[J]. 新疆大学学报(自然科学版), 2016, 33(1): 80-87
QIU Lei, LIU Huiling, LEI Jianlong. Principle explanation and performance analysis of jump point search algorithm[J]. Journal of Xinjiang University (Natural Science Edition), 2016, 33(1): 80-87
- [17] 张洪海, 李翰, 刘皞, 等. 城市区域物流无人机路径规划[J]. 交通运输系统工程与信息, 2020, 20(6): 22-29
ZHANG Honghai, LI Han, LIU Hao, et al. Path planning for logistics unmanned aerial vehicle in urban area[J].

Journal of Transportation Systems Engineering and Information Technology, 2020, 20(6): 22-29

- [18] 王秀红,刘雪豪,王永成.基于改进 A* 算法的仓储物流移动机器人任务调度和路径优化研究[J].工业工程, 2019, 22(6): 34-39

WANG Xiuhong, LIU Xuehao, WANG Yongcheng. A research on task scheduling and path planning of mobile robot in warehouse logistics based on improved A* algorithm [J]. Industrial Engineering Journal, 2019, 22(6): 34-39

- [19] 张庆,刘旭,彭力,等.融合 JPS 和改进 A* 算法的移动机器人路径规划[J].计算机科学与探索, 2021, 15

(11): 2233-2240

ZHANG Qing, LIU Xu, PENG Li, et al. Path planning for mobile robots based on JPS and improved A* algorithm [J]. Journal of Frontiers of Computer Science and Technology, 2021, 15(11): 2233-2240

- [20] 黄智榜,胡立坤,张宇,等.基于改进跳点搜索策略的安全路径研究[J].计算机工程与应用, 2021, 57(1): 56-61

HUANG Zhibang, HU Likun, ZHANG Yu, et al. Research on security path based on improved hop search strategy [J]. Computer Engineering and Applications, 2021, 57(1): 56-61

AGV path planning via integration of jump point search with bi-directional parallel ant colony algorithm

LIN Xinchuan¹

¹ College of Information Engineering, Fujian Business University, Fuzhou 350012, China

Abstract In this article, an improved algorithm that combines jump point search and bi-directional parallel ant colony search is proposed for static grid maps to address the slow convergence and easy trapping in local optima perplexed traditional ant colony algorithms for Automated Guided Vehicle (AGV) path planning. First, the actual research environment is modeled by gridization, and the improved jump point search algorithm is used to generate the initial suboptimal path for bi-directional search, providing a reference for the initial search direction of bi-directional ant colony search. Second, an improved transition probability heuristic function is used in the bi-directional parallel ant colony search process, which considers the avoidance of collision between AGV and obstacles when determining the next transition node. Meanwhile, by designing an information sharing mechanism and combining two fusion models of improved information increment and concentration, the global information concentration is shared and updated to better explore and optimize the path and ensure the connection of bi-directional paths. Finally, experimental results are compared with those of traditional ant colony algorithms to verify the improved algorithm's global search capability, efficiency and security.

Key words jump point search algorithm; ant colony algorithm; automated guided vehicle (AGV); path planning; bi-directional parallel